ACEEE Int. J. on Network Security , Vol. 5, No. 1, January 2014

Selfish Node Isolation & Incentivation using Progressive Thresholds

Dais John[1], Rosna P Haroon[2]
[1] Department of Computer Science and Engineering,
Ilahia College of Engineering and Technology, Muvattupuzha, Kerala, India
Email: daisjohnhere@gmail.com
[2] Department of Computer Science and Engineering,
Ilahia College of Engineering and Technology, Muvattupuzha, Kerala, India
Email: rosna.shihab@gmail.com

***Abstract*— The problems associated with selfish nodes in MANET are addressed by a collaborative watchdog approach which reduces the detection time for selfish nodes thereby improves the performance and accuracy of watchdogs[1]. In the related works they make use of credit based systems, reputation based mechanisms, pathrater and watchdog mechanism to detect such selfish nodes. In this paper we follow an approach of collaborative watchdog which reduces the detection time for selfish nodes and also involves the removal of such selfish nodes based on some progressively assessed thresholds. The threshold gives the nodes a chance to stop misbehaving before it is permanently deleted from the network. The node passes through several isolation processes before it is permanently removed. Another version of AODV protocol is used here which allows the simulation of selfish nodes in NS2 by adding or modifying log files in the protocol.**

***Index Terms*— Selfish nodes, Watchdogs, AODV.**

I. Introduction

A selfish node can be any node in the network which uses the network but doesn't cooperate in any kind of positive network activities. MANET is a kind of wireless ad-hoc network which is an autonomous collection of mobile nodes so the network topology may change rapidly and unpredictably over time. We can say that MANET is a self-configuring network of mobile routers connected by wireless links without any access point.

A MANET topology can also be defined as a dynamic (arbitrary) multi-hop graph G = (N, L), where N is a finite set of mobile nodes (MNs) and L is a set of edges which represent wireless links[2]. A link (i, j) $\in$ L exists if and only if the distance between two mobile nodes is less or equal than a fixed radius r as shown. This r represents the radio transmission range that depends on wireless channel characteristics including transmission power. Accordingly, the neighborhood of a node x is defined by the set of nodes that are inside a circle (assume that MNs are moving in a two-dimensional plane) with center at x and radius r, and it is denoted by:

$$N_r(x) = N_z = \{n_f \mid d(x,n_f)\} \leq r,\ x \neq n_f,\ \forall j \in N,\ j \leq |N|\} \quad (1)$$

Where x is an arbitrary node in graph G and d is a distance function.

Type of attacks that affect the routing protocol of ad-hoc network is called as routing description attacks such as black hole, grey hole and selfish node. A black hole attack absorbs the packet without forwarding them to the destination by falsely claiming a fresh route to the destination whereas grey-hole attack selectively drops some packets [3]. A selfish node uses the network but it doesn't cooperate with network and thereby saving battery life for its own communication. It doesn't intend to directly damage other nodes. One solution to such misbehaving nodes is to forward packets only through nodes that share a priori-trust relationship [4]. The main problem associated with such type of forwarding are more overhead, issues with key distribution etc. Sometimes untrusted nodes may behave well but creates some problems. An alternative solution for detecting such nodes is use of watchdogs. Also developed a threshold based algorithm for removal of such selfish nodes from the network by giving them a chance to come back to the network. Many properties of the nodes taken into account to decide when to remove nodes from the network. Nodes running on low battery are given a higher threshold than others to give justice to them. Similarly, nodes that are sending comparatively lesser amounts of data are also given a concession on the threshold in accordance with their data rates.

II. Related Works

Several solutions were proposed to the detection of selfish nodes in the network. A Bayesian watchdog [5] was introduced to improve the accuracy of detection. It is designed to be protocol independent, makes them compatible with all the routing protocols. They introduced two new mechanisms namely tolerance threshold and devaluation for moderating the false positives and the false negatives. The watchdog method detects misbehaving nodes. Suppose there exists a path from node S to D through intermediate nodes A, B, and C. Node A cannot transmit all the way to node C, but it can listen in on node B's traffic. Thus, when A transmits a packet for B to forward to C, A can often tell if B transmits the packet. If encryption is not performed separately for each link, which can be expensive, then a can also tell if B has tampered with the payload or the header. They implement the watchdog by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on.

DOI: 01.IJNS.5.1.13

If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. The watchdog technique has advantages and weaknesses.

The Problem is that a misbehaving node that can control its transmission power can circumvent the watchdog. A node could limit its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the true recipient [8]. This would require that the misbehaving node know the transmission power required to reach each of its neighbouring nodes. Only a node with malicious intent would behave in this manner selfish nodes have nothing to gain since battery power is wasted and overloaded nodes would not relieve any congestion by doing this

Paper [7] describes about two techniques that are used in ad-hoc network in presence of selfish nodes. They used both watchdog and pathrater mechanisms in their work. One of the main problem with this approach is that the pathrater works successfully only when the watchdog is active. They used DSR (dynamic source routing protocol) to implement their work. In this paper they analyse two possible extensions of to DSR to mitigate the effects of routing misbehaviour in ad-hoc networks-the watchdog and pathrater. They show that the two techniques increase throughput by 17% in a network with moderate mobility, while increasing the ratio of overhead transmission to data transmission from the standard routing protocol's 9% to 17%. During extreme mobility, watchdog and pathrater can increase network throughput by 27%, while increasing the percentage of overhead transmissions from 12% to 24%.

In [6], presents a logical survey on to detect the misbehaving nodes in the MANET using intrusion detection system and which detects the possible attacks in the network but the presence of collision makes them not work correctly and lead to wrong accusation.

In [13], uses a set of mobile agents that can move from one node to another within the network. It describes a method to reduce the network bandwidth consumption. Here the data analysis computation is performed at the location of intrusion. This technique requires much more time. They are mainly focused on the denial of service attack caused by the selfish nodes. The mobile ad-hoc networks suffer from several types of intrusions out of which, the denial of service attack by a selfish node is the one of them. The mobile agents travel through the network gathering vital information. This information is then processed by the mobile agent itself. The choice of threshold value is very important to help the detection of the attacker as early as possible.

The computation complexity of the mobile ad-hoc is kept minimum so that computation overhead can be reduced. In this paper they mainly focused on the DOS attack caused by selfish node by refusing the packet delivery to the neighbour node [9]. The computation overhead of our algorithm is much less as the computation is done by the mobile ad-hoc when the source node notices that the destination node does not response in correct time. The nodes are also free from performing the computation which increase the efficiency of each node thereby increasing the overall performance of the network.

Paper [11] presented a model to describe behaviour of an intermediary node in a route between a source and a destination, which is a collaborative point of its one-hop neighbours. This model is general enough to describe cooperation enforcement mechanism that have been proposed in literature in recent times, and it can be used to understand at what extent a node can be selfish, and how much can we pretend from it. From the investigations, it is found that model is able to regulate the selfishness based on residual energy. With higher energy, the node is able to contribute more cooperation and as well as more packet delivery ratio. Under steady state conditions, convergence of expected cooperation depends on the number of neighbours in the cluster. More neighbours in the cluster will bring more cooperation. Also, they study the impact of node misbehaviours and failures on network survivability, which is defined as the probabilistic k-connectivity of the network induced by active nodes.

Finally, Tehy showed that the network survivability turns out to be a function of the network properties (network size N, transmission range r, and initial density) and node behaviour distributions. As a conclusion, the impact of node behaviours (failures) on network survivability can be evaluated quantitatively from our analytical result, which can be further used as a guideline to design or deploy a survivable ad hoc network given a predefined survivability preference

One way of preventing selûshness in a MANET is a detection and exclusion mechanism. In paper [12], they focus on the detection phase and present diûerent kinds of sensors that can be used to ûnd selûsh nodes. In this paper we have presented a number of diûerent sensors that can detect diûerent kinds of selûsh nodes with a good conûdence as shown by our simulation results. If multiple sensors are active in parallel and a selûsh node is detected by a number of these sensors, then this is a good indication for excluding the node from the network.

One remaining problem with their current simulations is that all the thresholds need to be set manually in order to get good detection results. So in the future we will try to ûnd ways how these values can be set and adjusted automatically during operation. Possible candidates might be some kind of an adjustment algorithm or a self-learning system using neural networks.

Two techniques [10] namely reputation based technique and credit based technique are also used for the detection of selfish nodes in the network. In credit based schemes in order to perform network function successfully it offers incentives for the nodes. Two models called Packet purse model [PPM] and Packet trade model [PTM] are mainly used in this scheme. In PPM the source node needs to pay for the packet forwarding service. So the originator node loads it with

DOI: 01.IJNS.5.1.13

number of beans and requires more time also. In PTM model the total cost of packet forwarding is covered by the destination node. In reputation based scheme, it collectively detect the misbehaviour of a suspicious node and propagate this information throughout the network. Two models used by this scheme are watch dog model and pathrater. Pathrater calculates a path metric. It do so by averaging the node ratings in the path. Watchdog activities with frequent checking degrades the network performance by increasing the time of transmission due to continues check. In an average we can say that the existing system requires more detection time and more overhead. They discussed mainly about two techniques reputation based and credit based technique for detection of selfish nodes in MANET.

Two algorithms discussed here are based on the reputation based scheme and one is based on credit based scheme. The 2ACK scheme uses the reputation based approach to detect and mitigate the effect of misbehaving nodes in MANET. It serve as an add-on technique for routing scheme to detect routing misbehaviour and to moderate undesirable effect. The main idea is to send the two hop acknowledgement packets in opposite direction of the routing path. In order to reduce the additional routing overhead, only a fraction of the received data packets are acknowledged. Thus it detects the selfish nodes, eliminates them and chooses the other path for transmitting the data.

In reputation based approach, in addition to punishing the selfish nodes. And encouraging the cooperating nodes, there is second chance for the nodes which dropped a packet unwillingly. In this approach if the node is recognized to be a selfish node for the first time and punished, the cooperation coefficient of it can be increased if it changes its behaviour as a co-operator node. The third algorithm 'an auction based AODV protocol uses auctions for an ad-hoc network that consists of selfish nodes. It is based on incentivizing cooperation by balancing two different metrics: the residual energy and current currency level of the nodes in the network. It is implemented at the route level, rather than at each intermediate node, thus guaranteeing that a successful bid leads to an end-to-end transmission route.

In paper [12] they detect a large range of attacks on Dynamic Source Routing (DSR) protocol and the originator of the attack. They provide mechanism to inform other nodes of the system about the accused, provide a context aware inference scheme to blame the accused and malicious accuser without doubt. They have analysed the security threats in a DSR based co-operative ad-hoc network and presented a context aware inference scheme for source node to blame and rate an accused node conclusively. This happens by involving neighbourhood nodes into the inference mechanism. In the presence of a distributed certification authority this approach suits for also revealing the detected malicious node to nodes other than source and destination in ad-hoc community. This can be reached by adding digital signatures of each accusing node into the RREQ. Each node of the ad-hoc community, after receiving A's or (B's) claims, is then enabled to verify the culprit on its own by again starting the inference scheme for a certain Context. Nevertheless, such a rating-model is reasonable only when a malicious intermediate node is not enabled to join the ad-hoc community at a later point of time with a new address or new identifier.

DSR with the watchdog has the advantage that it can detect misbehaviour at the forwarding level and not just the link level [5]. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of any of the following:

1 Ambiguous collisions
2 Receiver collisions
3 Limited transmission power
4 False misbehaviour
5 Collusion
6 Partial dropping

DSR uses no periodic routing messages like AODV.

III. Problem Domain

This work mainly focus on the topic of Selfish Nodes within a Mobile Ad-Hoc Networks (MANET), MANETs have their design flaws and security concerns. One such issue is the existence of one or more selfish nodes within the network. Selfish nodes are nodes within the network that wish to conserve their own power, therefore they deny receiving packets from other nodes, while at the same time attempt to send packets of their own to its neighbours .

Selfish nodes can cause major concerns in a MANET, from dropping single packets to the point where no node can send any message, therefore taking the entire network offline [14]. Here a system developed on the basis of a predefined threshold, disables or permanently deletes selfish nodes from the network. Detection of the selfish nodes is essential for network performance. An alternative solution for the detection of such nodes is the use of watchdogs.

A way to reduce the detection time and to improve the accuracy of watchdogs is the collaborative approach. Sometimes the nodes act as selfish because of some other reasons like low battery life, less number of resources, battery power etc. Presence of such nodes in network will reduce overall network performance while routing so the removal of such nodes is essential. To achieve this a combined isolation and incentivation technique is used. Isolation methods are intended to keep the misbehaving nodes outside the network, excluding them from all kinds of communication. incentivation methods try to convince the selfish nodes to change their behaviour. Contact dissemination method is used to reduce the detection time for selfish nodes. In such method a node can spread information regarding selfish node to other nodes whenever a new contact occurs. The watchdog mechanism is used to detect the selfish nodes [1]. In [1] the network is modelled as a set of N wireless mobile nodes, with C collaborative nodes and S selfish nodes (N = C + S). It is assumed that the occurrence of contacts between two nodes follows a poison distribution ›. They modelled the performance using a Continuous Markov chain with two parameters to indicate the degree of collaboration and

DOI: 01.IJNS.5.1.13

detection of the watchdog. Numerical results shows that a collaborative watchdog can reduce the overall detection time with a reduced cost in term of message overhead. This reduction is very important when the watchdog detection effectiveness is low. The isolation and incentivation method comprises of two algorithms mainly, for their proper working.

IV. Problem Definition

In a MANET, nodes can freely move around while communicating with each other. These networks may underperform in the presence of nodes with a selfish behaviour, particularly when operating under energy constraints. A selfish node will typically not cooperate in the transmission of packets, seriously affecting network performance. Although less frequent, nodes may also fail to cooperate either intentionally (a malicious behaviour) or due to faulty software or hardware [13]. Therefore, detecting these nodes is essential for network performance. The effects of misbehaviour have been shown to dramatically decrease network performance. Depending on the proportion of misbehaving nodes and their strategies, network throughput decrease, packet loss, denial of service, and network partitioning can result. These detrimental effects of misbehaviour can endanger the entire network. To overcome these problems removal of such selfish nodes is essential.

The algorithm proposed in this paper is detection and removal of selfish nodes. The main objective is to identify and isolate selfish nodes from the network. Through successful isolation, the MANET performance will be increased and will become more reliable. The main two parameters considered here are the battery power of the node and the traffic through that node. Nodes with a low battery power should be given some consideration and a higher threshold. Also, nodes doing more send-receive should have a lower threshold since they use the network more. At the end an improvement can be done to use the average battery power than the current value.

V. Problem Statement

Selfish node isolation and incentivation by progressive threshold based on traffic and battery power of the nodes.

VI. Proposed System

This work uses a collaborative watchdog mechanism which is a scheme to identify misbehaving nodes by monitoring next node's transmission. A collaborative watchdog approach reduces the detection time for selfish nodes thereby improving the network performance and also the accuracy of the watchdogs. Here when a node forwards a packet the watchdog will check whether the next node in the route also forwards the packet or not. To achieve this the watchdog mechanism continuously listening to next node's transmission with information collected about the neighborhood behavior the watchdog can decide whether the node acting as selfish or not. Once a node is detected to be selfish this watchdog can spread this information to other nodes when a new contact occurs.

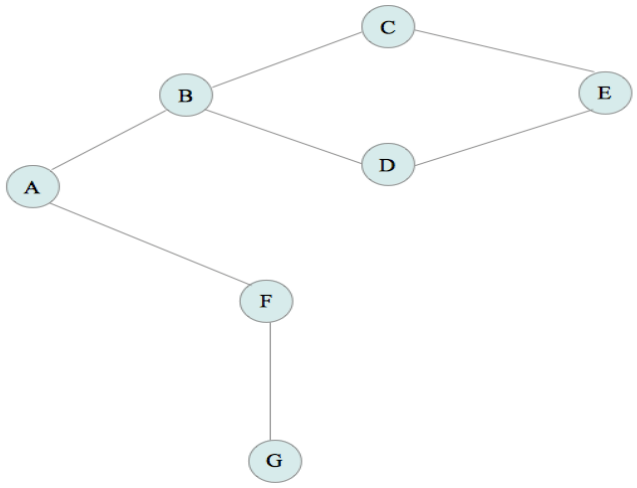

Figure 1. Watchdog Mechanism

In fig.1 suppose A wants to send a packet to E and the available routes to E are A_B_C_E and A_B_D_E. Here B can forward packets to both C and D. but d act as selfish and it drops all the packets received from B. When D drops packet watchdog in B can detect it and mark D as selfish. Whenever a new contact occurs to B it can pass this information to the new contact.

AODV is an ad-hoc on demand distance vector routing protocol [15]. It is a method for routing messages between mobile nodes. AODV allows messages to pass through their neighbors to the destination node with which they cannot communicate directly. The protocol creates a path to the destination node on demand. Whenever a node needs to forward a packet to some destination, it first checks its routing table to determine whether a route to the destination is already available or not. If \a path is already available, it forwards the packet to next-hop node. If a route is not available, then the node initiates a route discovery process. The protocol is able to handle changes in routes in presence of errors in existing routes. AODV has less average traffic throughput - which means that less traffic is forwarded on each hop and therefore there is less probability of collision.

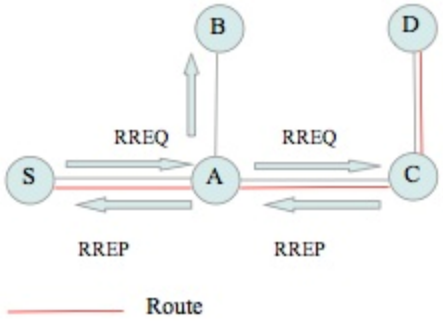

Figure 2. Working of AODV protocol

Once a node detected as selfish, we can develop a continuous monitoring system for watching the selfish node. In this paper, we propose a system which on the basis of a

DOI: 01.IJNS.5.1.13

predefined threshold, disables or permanently deletes selfish nodes from the network.

AODV is also on-demand routing protocol, hence the route is established only when it is required by the source for transmission of data packets. AODV uses destination sequence number (DestSeqNum) to identify the most recent path to the destination. A node updates its path information only if the DestSeqNum of th current packet received is greater than the last DestSeqNum stored at the node.

The DSR uses source routing in which a data packet carries the complete path to be traversed, whereas in AODV the source node and intermediate nodes store the next-hop information. DSR with the watchdog has the advantage that it can detect misbehavior at the forwarding level and not just the link level. It's weaknesses are that it might not detect a misbehaving node in the presence of: ambiguous collisions, limited power etc.

In the proposed system, the routing table stores two counters for each node namely drop_count and warning_count. We also define three values: drop_threshold, warning_threshold and block_duration..Whenever a node is detected to selfishly drop a packet, the drop_count for that node is incremented. Once the drop_count reaches the drop_threshold, we will mark the node as selfish, increment warning_count and also mark all routes that contain this node as disabled. The current time is also stored in the table to control how long the node is to be blocked. Thus we temporarily disable the node and all traffic through the node in the network. This is called isolation of the node.

VII. Mathematical Formulation

Consider the battery power of the node to decide the drop threshold/warning threshold. Nodes with a low battery power should be given some consideration and a higher threshold i.e. threshold will be inversely proportional to the battery power (in %) by some constant a.

In mathematics, two variables are proportional if a change in one is always accompanied by a change in the other, and if the changes are always related by use of a constant. The constant is called the coefficient of proportionality or proportionality constant. Formally, two variables are inversely proportional (also called varying inversely, if one of the variables is directly proportional with the multiplicative inverse (reciprocal) of the other, or equivalently if their product is a constant. It follows that the variable y is inversely proportional to the variable x if there exists a non-zero constant k such that

$$Y = K / X \quad (2)$$

$$\text{dropThreshold} = \text{baseDropThreshold} * a / (\text{Power}) \quad (3)$$

Here our Y is the dropThreshold and X is the battery power.

We need to update the protocol to send the battery power also as a header with each packet.

Another parameter to use is the amount of traffic. Nodes doing more send-receive should have a lower threshold since they use the network more. That is the Y value is the dropThreshold and X value is the traffic rate.

$$\text{dropThreshold} = \text{baseDropThreshold} * b / (\text{Traffic rate}) \quad (4)$$

The same can be applied for warning threshold also, if required.

After doing the above, we can do an improvement at the end to use the average battery power than the current value. If the power is constantly reducing, we will give a higher threshold. This can be done by keeping a weighted average of the battery power over time. A field in the routing table will contain the current average (avg). For the first packet it will be the value in the header. For the next packet we will update it to:

$$\text{Avgpower} = (\text{avgpower} * w) + \text{currentBatteryPower} * (1-w) \quad (5)$$

w is the weight factor . The weight factor w will be less than 0.5 so the current power will have a higher influence on the average compared to taking the mean. This ensures that historical values do not overwrite the recent trend.

VIII. Algorithm

A. Algorithm 1: For Isolating selfish nodes

1: Start

2: Initialize two counters for each nodes in the routing table: drop_count, warning count, current_time to a constant value

3: Initialize three variables: drop_threshold, warning_threshold and block_duration to some predefined value,

4: Whenever a node is detected to selfishly drop a packet increment Drop_count by one.

5: When Drop_count reache Drop_threshold set a flag for that node i.

6: Increment warning_count.

7: Set a flag for that node as disable.

8: Store current time to the variable current_time.

Whenever the routing table is checked for forwarding a packet, the sending node first checks if the route is marked as disabled. If it is disabled, it checks the timestamp in the table to check if a time of block_duration has elapsed since the node was blocked. If not, the route is not considered. However, If block_duration has already elapsed, the selfish flag is cleared, the route is re-enabled and the route is used. The drop_count and timestamp are also cleared at this point. This gives the node another chance to stop misbehaving and comply with the rules. The technique is called incentivation.

B. Algorithm 2: for incentivizing selfish nodes

1. Start

2. Each node ni checks the flag of neighboring nodes to decide whether it is disabled or not. IF yes go to step 3 else ignore the route.

3. Checks the timestamp against block
_duration to decide whether it is disabled or not, if yes go to step 4, and if not the route is not considered.

DOI: 01.IJNS.5.1.13

4. Clear the flag, set the flag as enabled, Clear Drop_count and timestamp.Each time the warning_count is incremented, it is checked against warning_threshold to see if the node has been warned more than the defined limit. If yes, then the non-cooperating node is permanently removed from the topology and all routes involving the node are cleared. The node is also added to a blacklist to prevent it from joining the network again.

IX. NS2 Simulation

According to Shannon simulation is "the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system. Ns2 is a time driven simulation tool which enables the simulation and analysis of packet sending detection and removal of selfish nodes in the MANET. The Simulation of wireless network functions and protocols such as TCP and routing algorithms such as AODV can be used. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors

A. Data Model

The data model is created to list set of inputs that are used in various stages of the work. The main inputs considered at various levels are listed.

The main inputs, undergoing processes respective outputs etc. of this work are listed in the following table.

The data packets send are monitored using collaborative watchdog approach. A method in aodv.cc will overhear the packet's transmission for checking whether the node forwarding the packet or not. The watchdog mechanism written in watchdog.cc file will find the drop count of selfish node. The trace file generated through the trace function given by the ns2 package will trace each node's transmission. The main inputs considered are:

1. Stream of packets through watchdog ()
2. Stream of packets through checkmal ()
3. Trace file to get the throughput graph.

The data model of this work is given in table I.

Table I. Data Model

NO	INPUT	PROCESS	TOOL	OUTPUT	REMARKS
1	Stream of packets	Aodv.cc	Tap()	Overhearing packet transmission, checking whether the node forwarding the packet or not. .	Aodv.cc is modified
2	Stream of packets	Aodv.cc	Chkmal()	Finding the drop count and warning count of malicious node	Aodv.cc is modified
3	Stream of packets	Watchdog.cc	Watchdog()	Watching each packet flow, maintaince the packet queue and neighbor nodes.	New code watchdog.cc is added
4	Ttracefile	Namtrace	Trace()	Trace each node's transmission	Trace file is automatically generated for each execution

B. Process Description

After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network AniMator) and XGraph are used. . NS2 can be invoked by executing the following statement from the shell environment:

ns [<file>] [<args>] where<file> and <args> are optional input argument. NAM trace is records simulation detail in a text file, and uses the text file the play back the simulation using animation. NAM trace is activated by the command "$ns namtrace-all $file", where ns is the Simulator handle and file is a handle associated with the file (e.g., out.nam in the above example) which stores the NAM trace information. After obtaining a NAM trace file, the animation can be initiated directly at the command prompt through the following command: namfilename.nam

AWK, the general-purpose programming language used for processing of text files. AWK processes an input file specified in an AWK script which can be can be specified at the command prompt or in a file. The Xgraph tool that allows to plot the results of the simulation in the form of curves also used here to get the output accurate. Throughput is calculated as per the output generated and plotted it with the help of Xgraph.

C. Result

The graph shows the performance analysis of the work. The X-axis indicates the time event and the Y-axis indicates the throughput calculated at each node. From this it is clear that the node throughput decreases as the node starts dropping.

The graph shows the performance analysis of the work. The X-axis indicates the time event and the Y-axis indicates the throughput calculated at each node. From this it is clear that the node throughput decreases as the node starts dropping. Calculating Packet delivery ratio is calculated by dividing the Packet data sent by time in second.

X. Comparative Study

A. Existing System

1. Detection only
2. Used isolation only
3. More detection time
4. More overhead
5. Nodes may blocked permanently
6. Less accuracy

B. Proposed System

1. Both detection and removal of selfish nodes

DOI: 01.IJNS.5.1.13

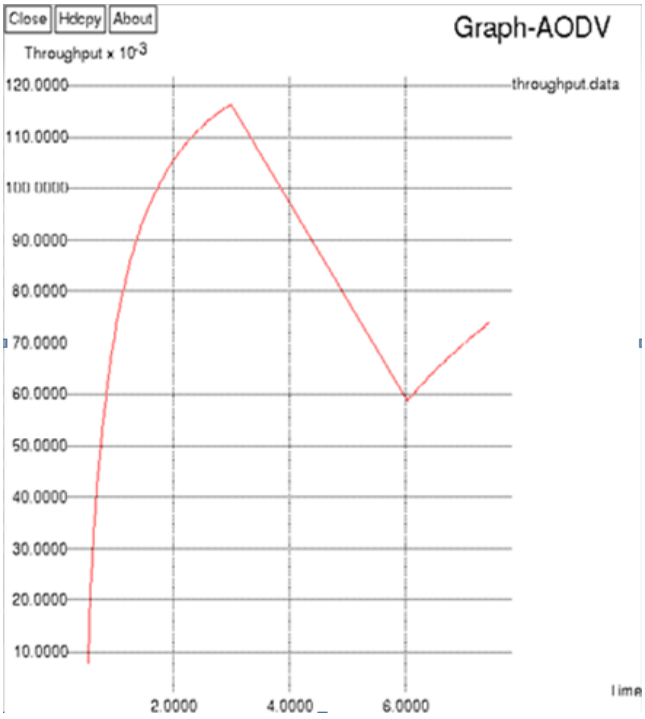

Fig.3 Throughput calculation

2. Both isolation and incentivation used
3. Less amount of detection time
4. Less overhead
5. Use of two thresholds so node blocked inadvertently
6. More accuracy

XI. Conclusion

The nodes are deleted only after having the node several chances to perform. The assumption is that existence of such a selfish node causes the following problems in a MANET such as:

1. Decreases network performance
2. Drops all the packets without forwarding
3. Never receives anything
4. Decrease the network accuracy
5. Cost of each node is minimum//reason
6. Parameter passing is limited//reason
7. Collaborative watchdog reduces detection time// reason

The advantage of permanent removal of selfish nodes are the following:

1. Having two sets of thresholds ensures that no node is blocked in advertently
2. The first threshold makes sure unexpected packet drops does not cause a node to be blocked.
3. The second threshold gives the node a second chance to stop misbehaving before it's permanently deleted from the network
4. The key goal of this work is to reduce the overhead of handling selfish nodes and increased accuracy
5. It involves both detection and removal of selfish nodes.
6. Less amount of detection time required.
7. Less overhead
8. Has two thresholds so no node is blocked inadvertently
9. More accuracy

A collaborative watchdog approach for detecting selfish nodes and an efficient threshold based mechanism for removal of such node from the network has been proposed in this paper. The proposed system uses two techniques namely isolation and incentivation to warn and control misbehaving nodes. Having two sets of thresholds ensures that no node is blocked inadvertently. The first threshold makes sure unexpected packet drops does not cause a node to be blocked. The count could also be reset after a period of no drop reports for better results.

The second threshold gives the node a second chance to stop misbehaving before it's permanently deleted from the network. The key goal of this work is to reduce the overhead of handling selfish nodes and increased accuracy.

XII. Future Work

Experiment has been done for count of 10 nodes in manet. As part of the sensitivity analysis the range of nodes can be incremented for a comparative study of computational complexity, drop_Count, Drop_thrshold, Warning_threshold.

Acknowledgement

The authors wish to thank the Management and Principal and Head of the Department (CSE) of Ilahia College of Engineering and Technology for their support and help in completing this work.

References

[1] Enrique Herna ìndez-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni, "improving selfish node detection in MANET using a collaborative watchdog" IEEE COMMUNICATIONS LETTERS, VOL. 16, NO. 5, MAY 2012

[2] Anuj K. Gupta, Member, IACSIT, Dr. Harsh Sadawarti, Dr. Anil K. Verm " Performance analysis of AODV, DSR & TORA Routing Protocols" IACSIT International Journal of Engineering and Technology, Vol.2, No.2, April 2010 ISSN: 1793-8236.

[3] Surana K A, Rathi S B, Thosar T P, Snehal Mehatre "Securing blackhole attack in rouing protocol AODV in MANET with watchdog mechanism." World Research Journal of Computer Architecture ISSN: 2278-8514 & E-ISSN: 2278-8522, Volume 1, Issue 1, 2012, pp.-19-23.

[4] T.V.P.Sundararajan1, Dr.A.Shanmugam2 "Modeling the Behavior of Selfish Forwarding Nodes to Stimulate Cooperation in MANET" International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010

[5] M. Annie Sharmila1, Dr.G. Murugaboopath" contact

DOI: 01.IJNS.5.1.13

dissemination based collaborative watchdog approach to improve selfish node detection in manets" International Journal of Emerging Technology and Advanced Engineering. Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013).

[6] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the uselfusness of watchdogs for intrusion detection in VANETs," in 2010 ICC Workshop on Vehicular Networking and Applications.

[7] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proc. 2000 MobiCom, pp. 255–265.

[8] S.Marti , T.J.Giuli , K.Lai , and , M.Baker , "A Performance and analysis of Misbehaving node in Manet using Detection system" in Proc.2009 MobiCom,pp.255-265

[9] Debdutta Barman Roy1 and Rituparna Chaki "MADSN: Mobile Agent Based Detection of Selfish Node in MANET" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 4, August 2011

[10] Dipali Koshti, Supriya Kamoji "Comparative study of Techniques used for Detection of Selfish Nodes in Mobile Ad hoc Networks" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-4, September 2011

[11] Pankaj Sharma,Yogendra Kumar Jain "TRUST BASED SECURE AODV IN MANET" Volume 3, No. 6, June 2012 Journal of Global Research in Computer Science

[12] Krishna Paul and Dirk westoff "Context aware detection of selfish nodes in DSR based ad-hoc networks"

[13] T.V.P.Sundararajan 1, Dr.A.Shanmugam "Modeling the Behavior of Selfish Forwarding Nodes to Stimulate Cooperation in MANET" International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010

[14] Frank Kargl, Andreas Klenk, Stefan Schlott, and Michael Weber "Advanced Detection of Selûsh or Malicious Nodes in Ad hoc Networks"

[15] Anuj K. Gupta, Member, IACSIT, Dr. Harsh Sadawarti, Dr. Anil K. Verma "Performance analysis of AODV, DSR & TORA Routing Protocols" IACSIT International Journal of Engineering and Technology, Vol.2, No.2, April 2010 ISSN: 1793-823.

DOI: 01.IJNS.5.1.13